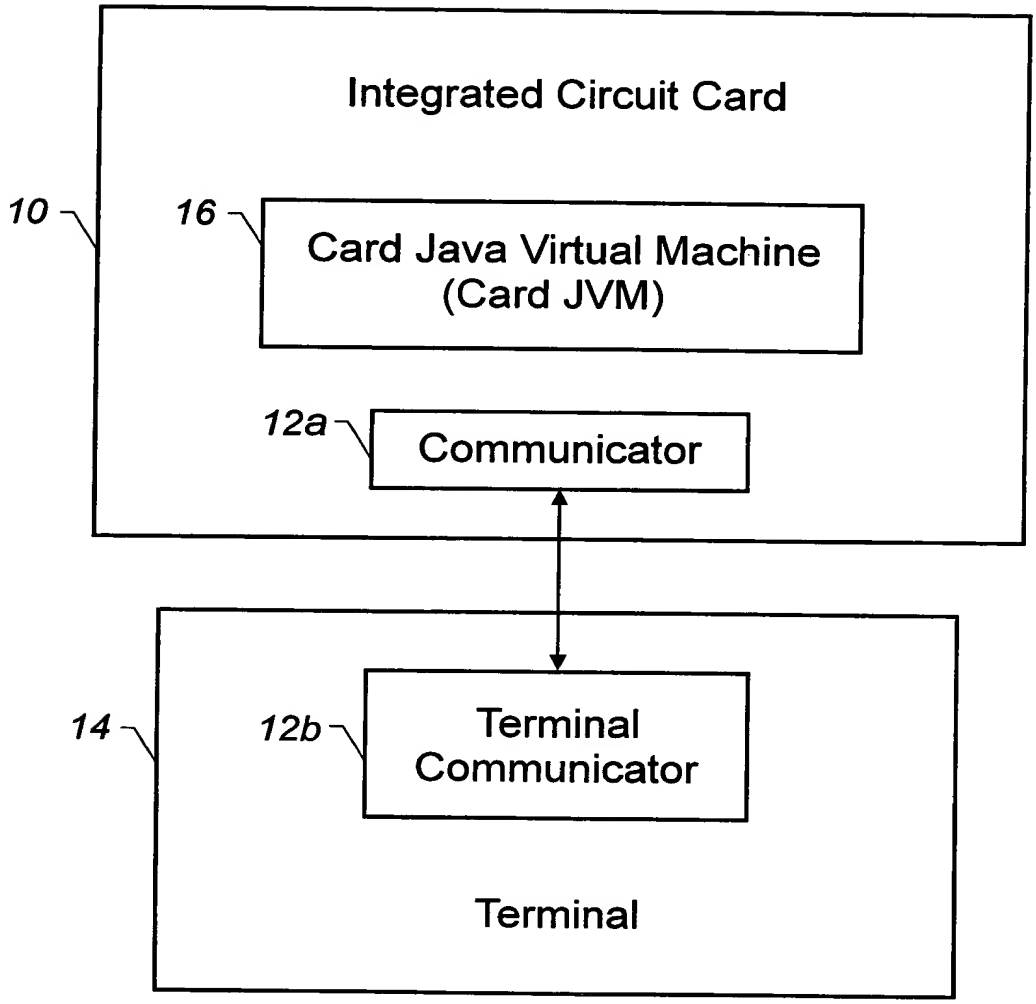

Integrated Circuit Card
10
16
Card Java Virtual Machine
(Card JVM)
12a
Communicator
14
12b
Terminal
Communicator
Terminal

FIGURE 1

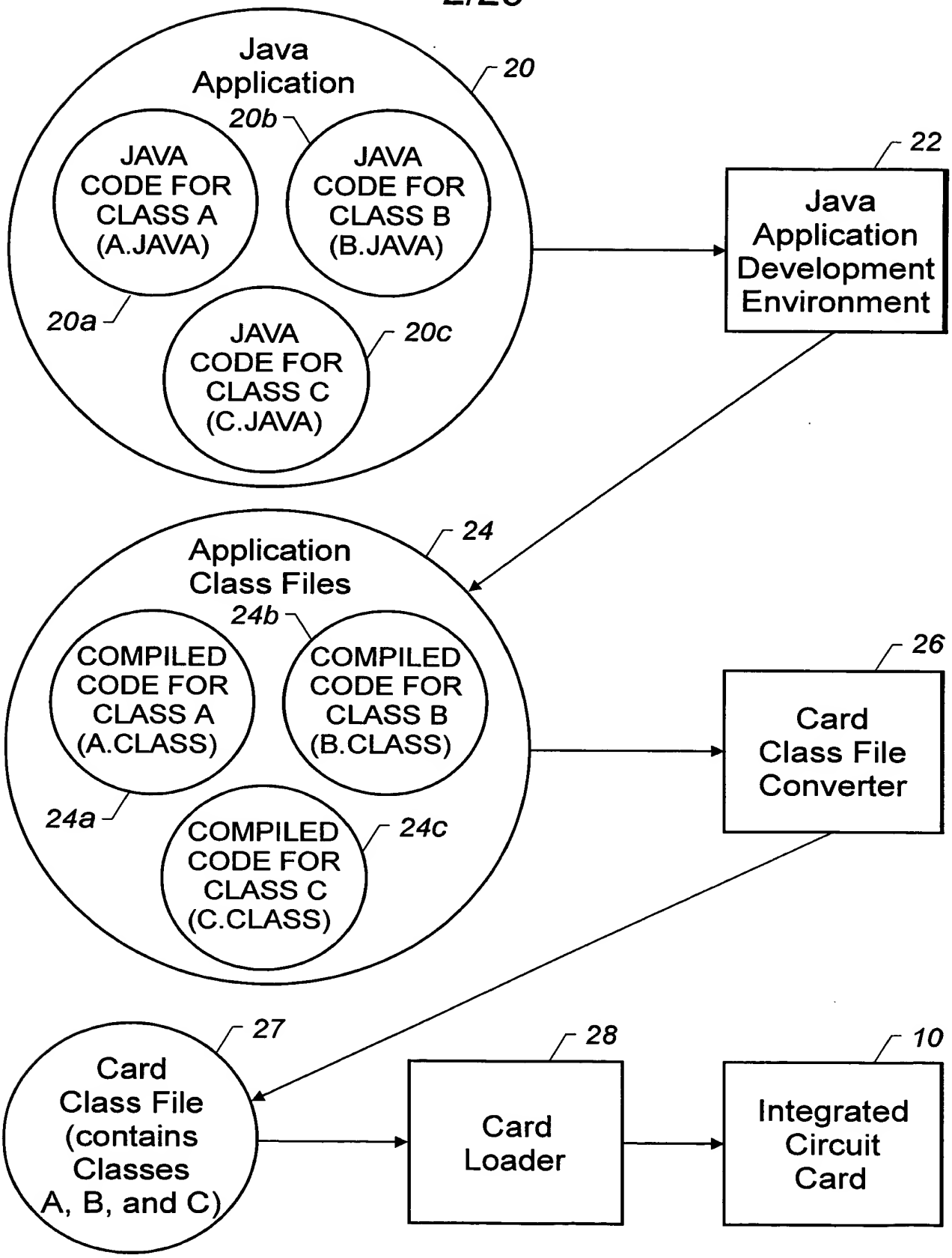

Java Application
20
20b
JAVA CODE FOR CLASS A (A.JAVA)
JAVA CODE FOR CLASS B (B.JAVA)
20a
JAVA CODE FOR CLASS C (C.JAVA)
20c
22
Java Application Development Environment
Application Class Files
24
24b
COMPILED CODE FOR CLASS A (A.CLASS)
COMPILED CODE FOR CLASS B (B.CLASS)
24a
COMPILED CODE FOR CLASS C (C.CLASS)
24c
26
Card Class File Converter
27
Card Class File (contains Classes A, B, and C)
28
Card Loader
10
Integrated Circuit Card

FIGURE 2

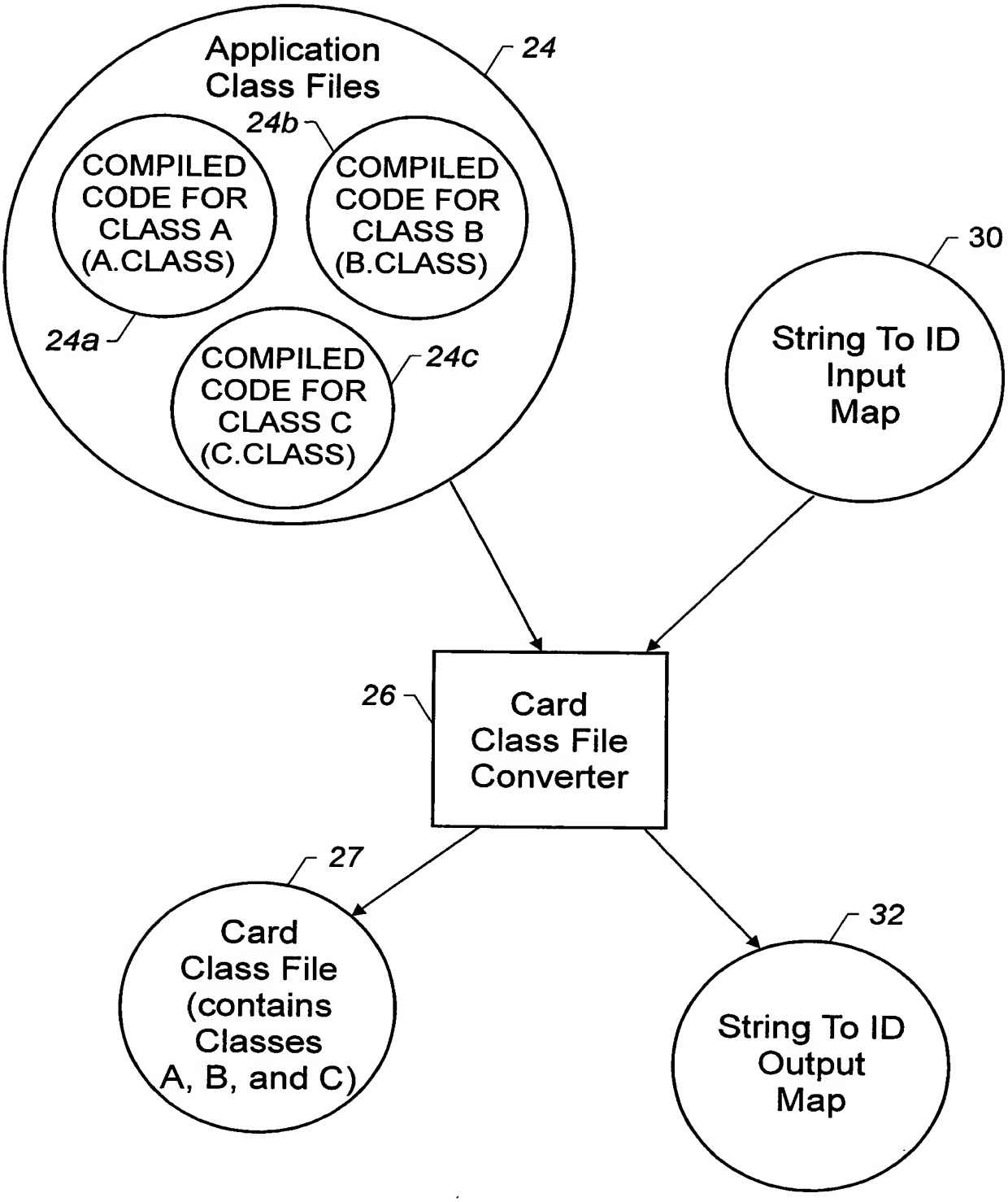

Application Class Files
24
24b
COMPILED CODE FOR CLASS A (A.CLASS)
COMPILED CODE FOR CLASS B (B.CLASS)
24a
COMPILED CODE FOR CLASS C (C.CLASS)
24c
30
String To ID Input Map
26
Card Class File Converter
27
Card Class File (contains Classes A, B, and C)
32
String To ID Output Map

FIGURE 3

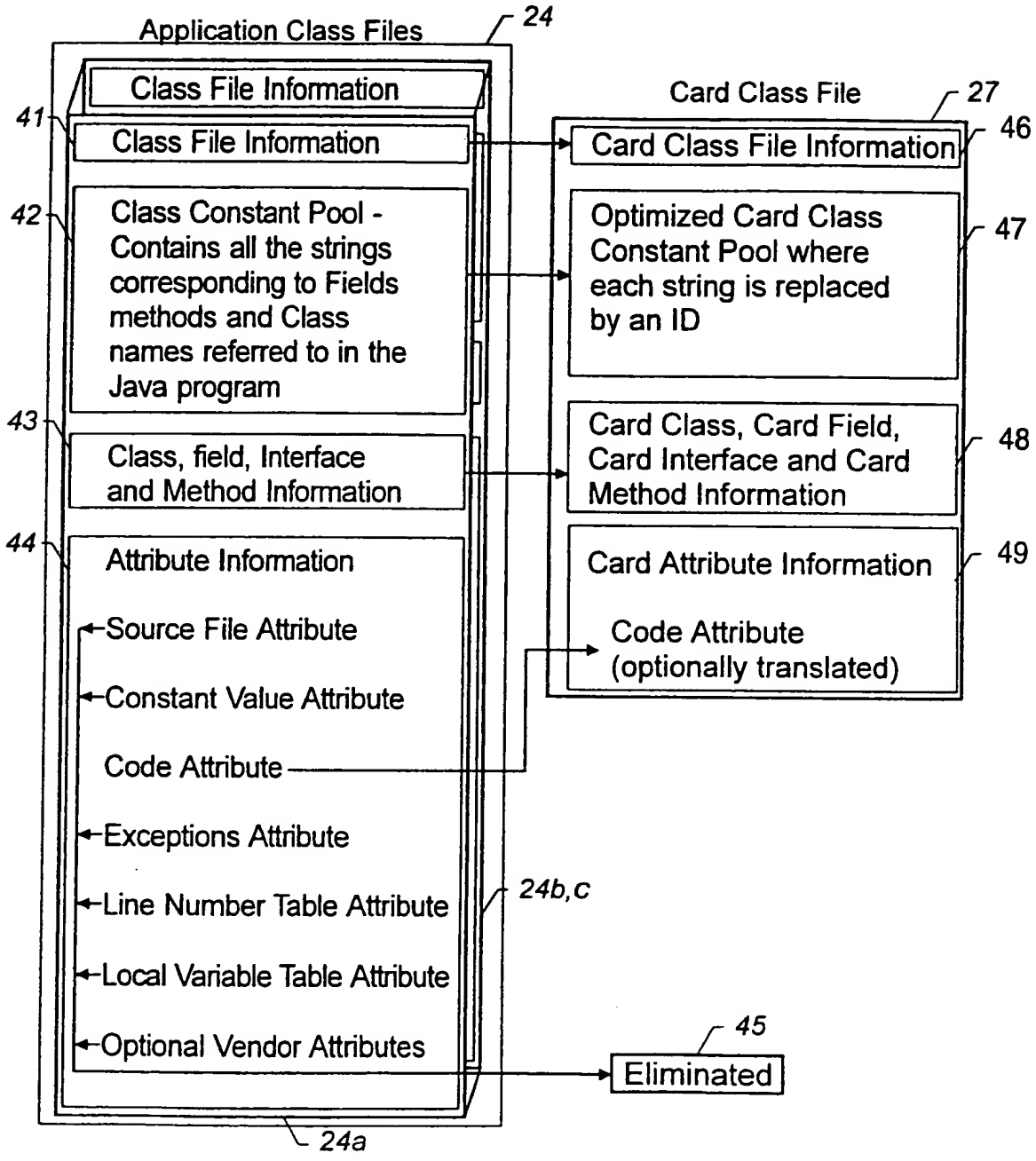

Application Class Files
24
Class File Information
41
Class File Information
42
Class Constant Pool - Contains all the strings corresponding to Fields methods and Class names referred to in the Java program
43
Class, field, Interface and Method Information
44
Attribute Information
Source File Attribute
Constant Value Attribute
Code Attribute
Exceptions Attribute
Line Number Table Attribute
Local Variable Table Attribute
Optional Vendor Attributes
24b,c
24a
Card Class File
27
Card Class File Information
46
Optimized Card Class Constant Pool where each string is replaced by an ID
47
Card Class, Card Field, Card Interface and Card Method Information
48
Card Attribute Information
49
Code Attribute (optionally translated)
45
Eliminated

FIGURE 4

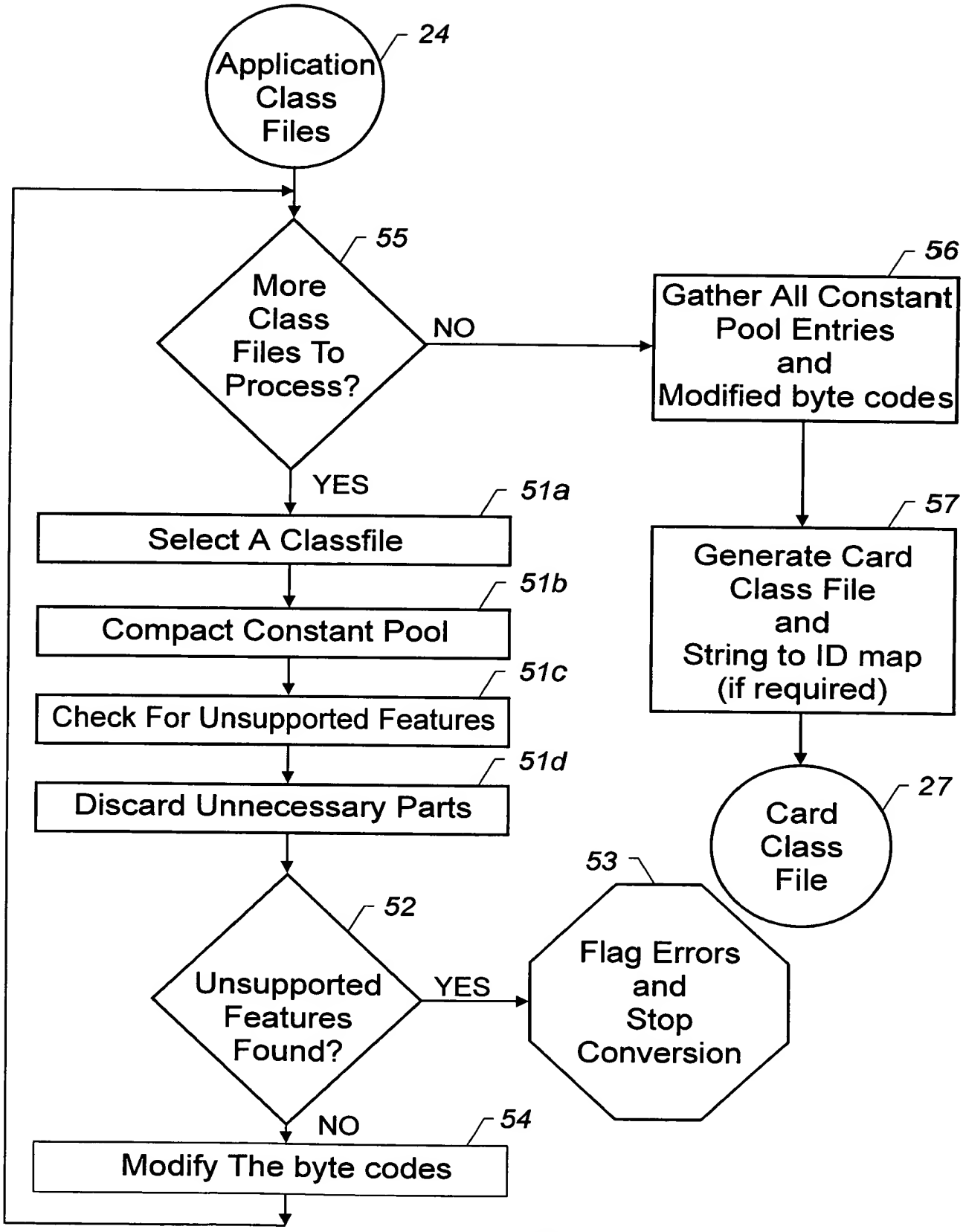

Application Class Files
24
More Class Files To Process?
55
NO
YES
Gather All Constant Pool Entries and Modified byte codes
56
Select A Classfile
51a
Compact Constant Pool
51b
Check For Unsupported Features
51c
Discard Unnecessary Parts
51d
Generate Card Class File and String to ID map (if required)
57
Card Class File
27
Unsupported Features Found?
52
YES
Flag Errors and Stop Conversion
53
NO
Modify The byte codes
54

FIGURE 5

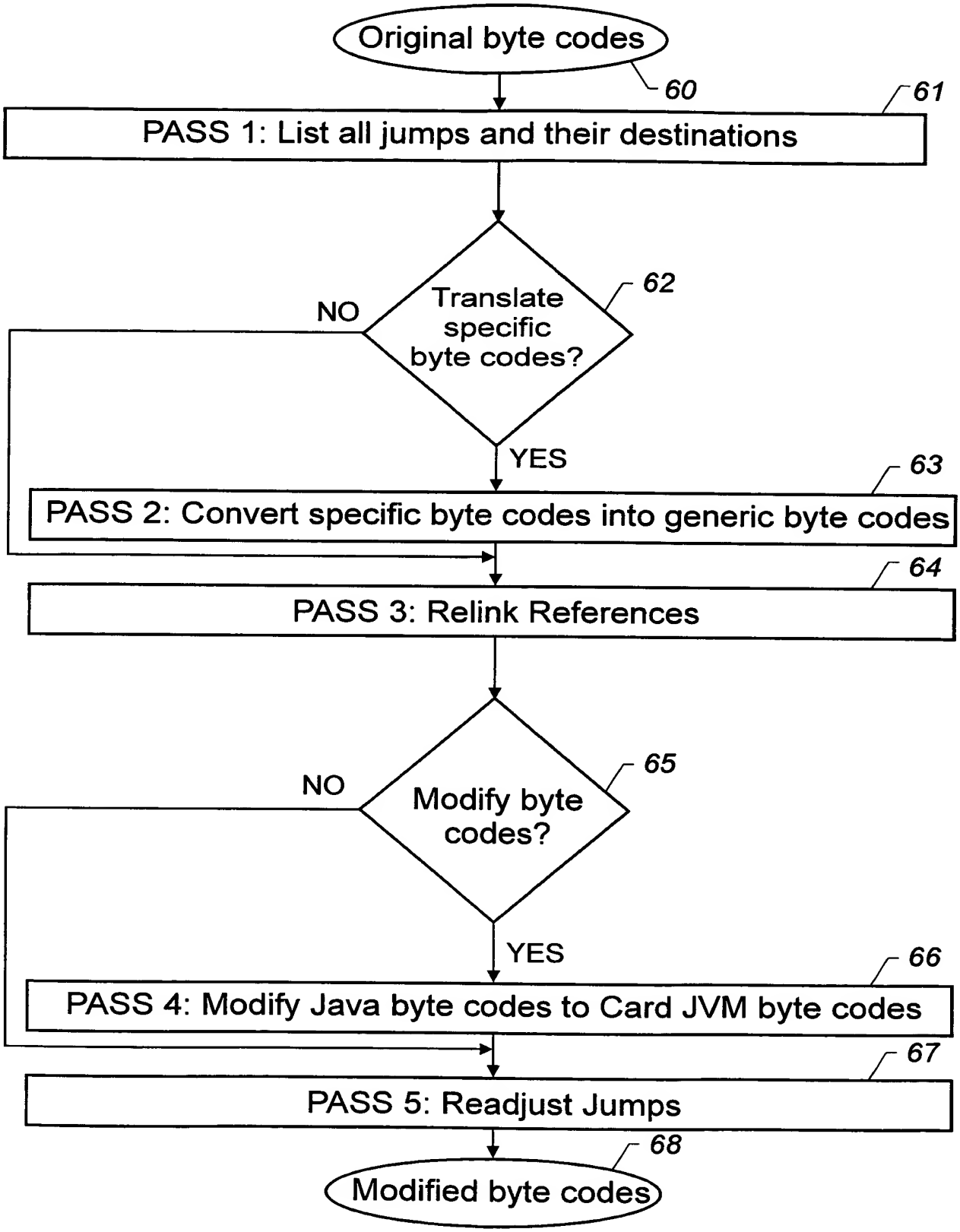

Original byte codes
60
61
PASS 1: List all jumps and their destinations
Translate specific byte codes?
62
NO
YES
63
PASS 2: Convert specific byte codes into generic byte codes
64
PASS 3: Relink References
Modify byte codes?
65
NO
YES
66
PASS 4: Modify Java byte codes to Card JVM byte codes
67
PASS 5: Readjust Jumps
68
Modified byte codes

FIGURE 6

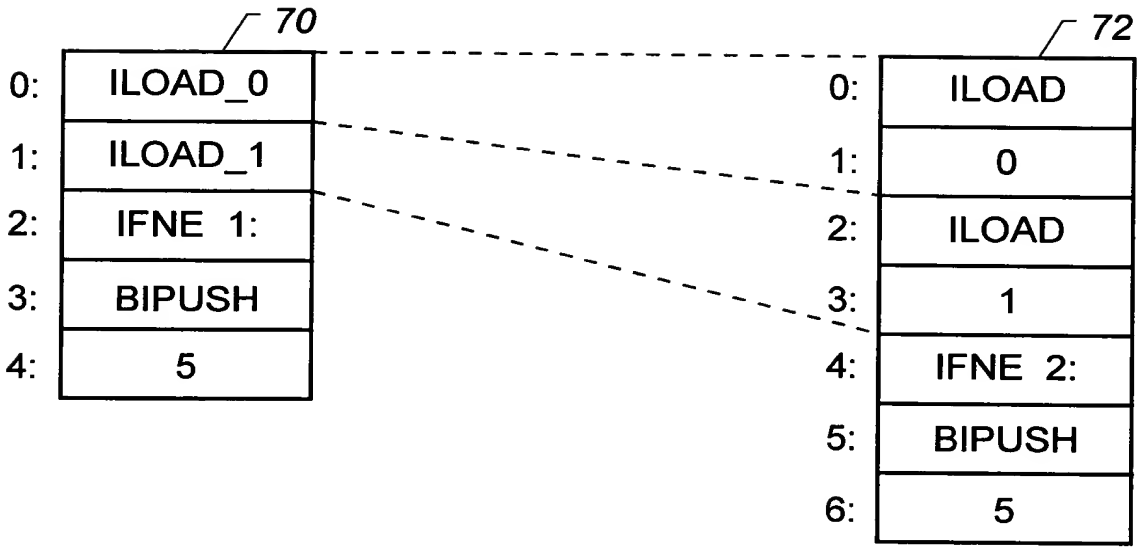
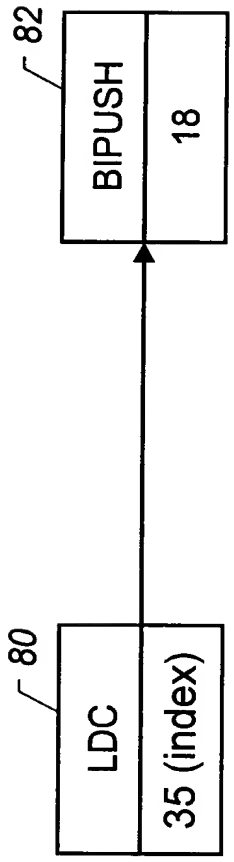
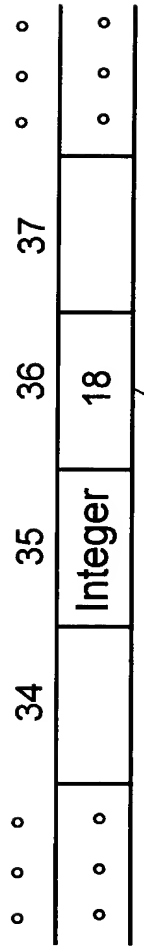

70
0: ILOAD_0
1: ILOAD_1
2: IFNE 1:
3: BIPUSH
4: 5
72
0: ILOAD
1: 0
2: ILOAD
3: 1
4: IFNE 2:
5: BIPUSH
6: 5

FIGURE 7





Class file Constant Pool

FIGURE 8

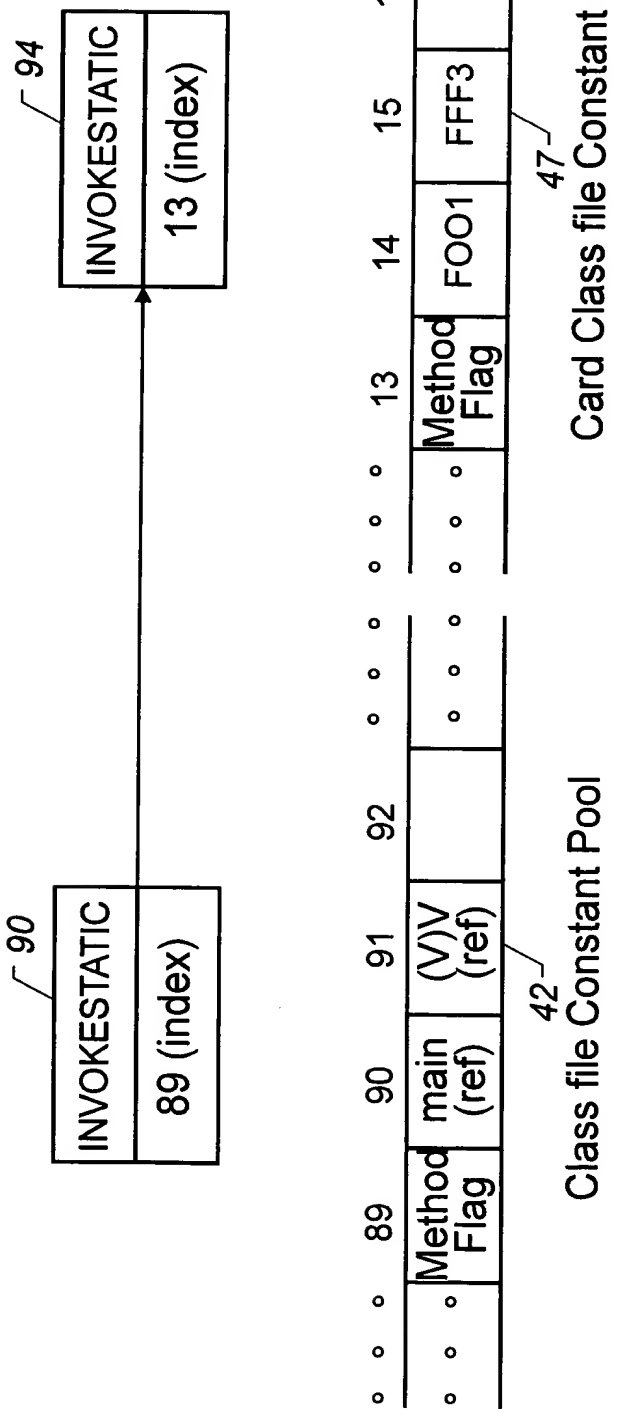

94
INVOKESTATIC
13 (index)
90
INVOKESTATIC
89 (index)
89
90
91
92
13
14
15
16
Method Flag
main (ref)
(V)V (ref)
Method Flag
FOO1
FFF3
42
Class file Constant Pool
47
Card Class file Constant Pool

FIGURE 9

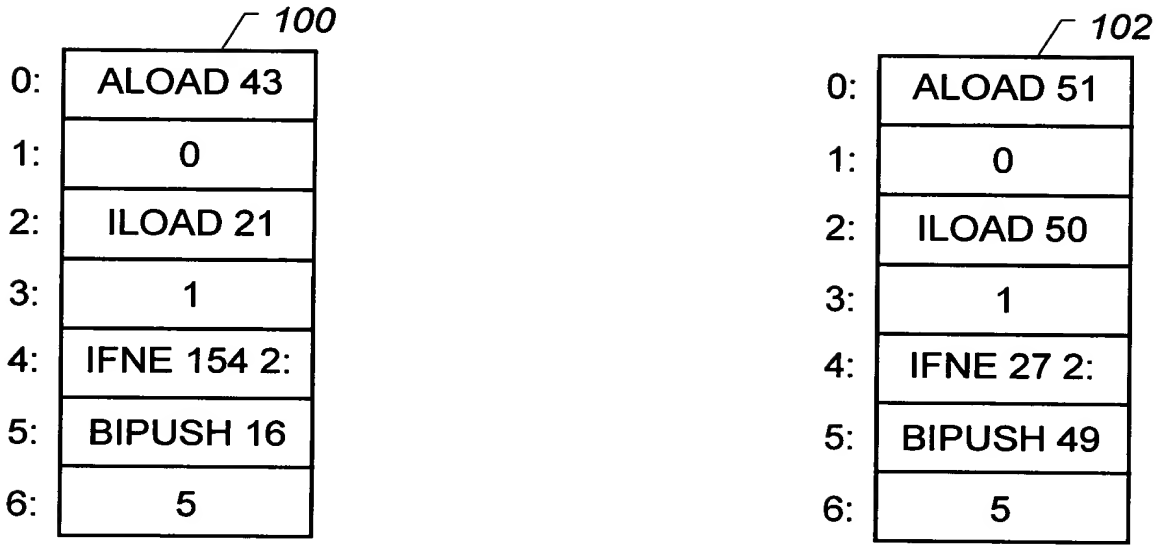

100
0: ALOAD 43
1: 0
2: ILOAD 21
3: 1
4: IFNE 154 2:
5: BIPUSH 16
6: 5
102
0: ALOAD 51
1: 0
2: ILOAD 50
3: 1
4: IFNE 27 2:
5: BIPUSH 49
6: 5

FIGURE 10

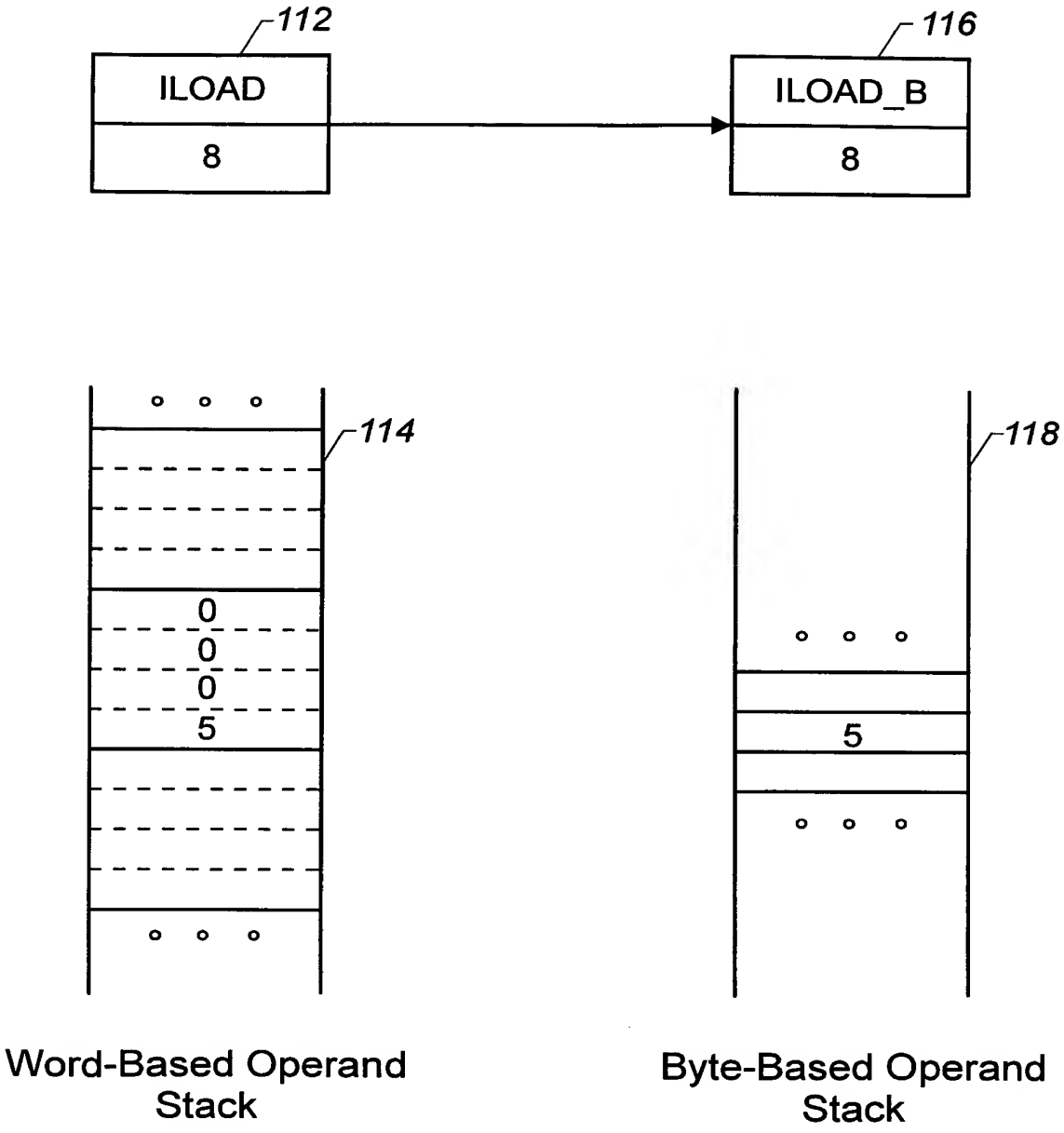

112
ILOAD
8
116
ILOAD_B
8
114
0
0
0
5
118
5
Word-Based Operand Stack
Byte-Based Operand Stack

FIGURE 11

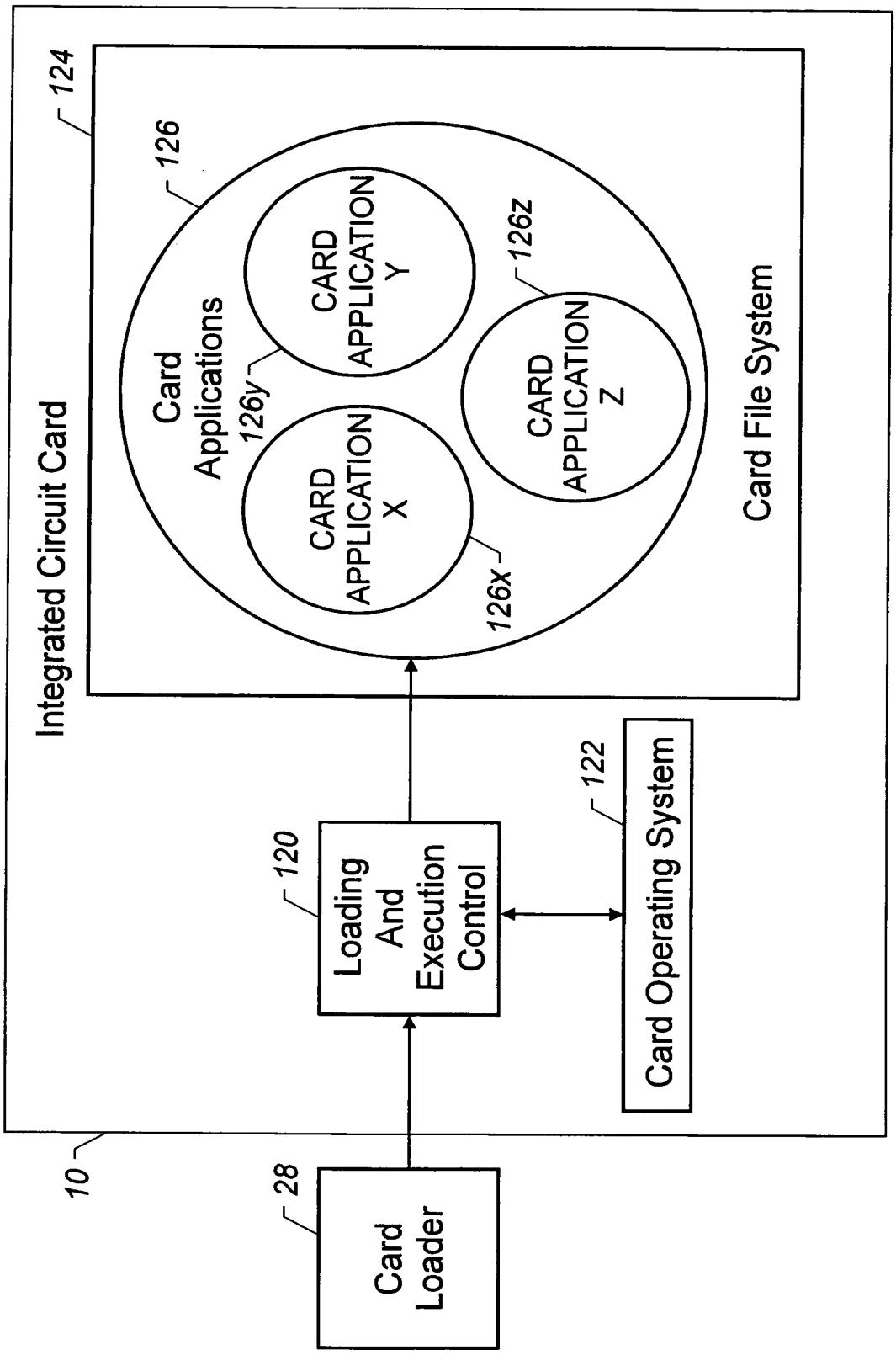

10
Integrated Circuit Card
124
126
Card Applications
126y
CARD APPLICATION Y
126z
CARD APPLICATION Z
CARD APPLICATION X
126x
Card File System
120
Loading And Execution Control
122
Card Operating System
28
Card Loader

FIGURE 12

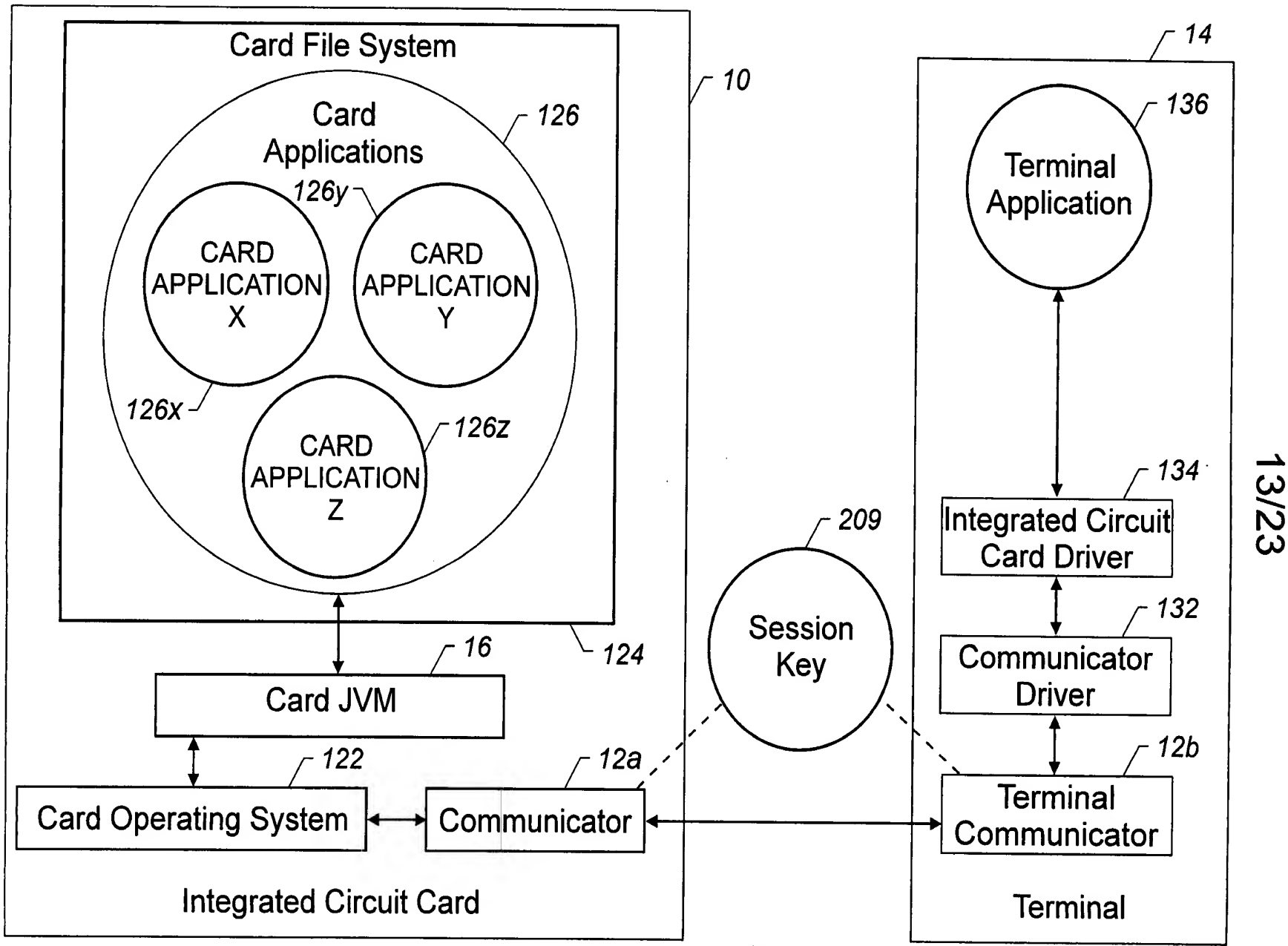

Card File System
10
Card Applications
126
126y
CARD APPLICATION X
CARD APPLICATION Y
126x
CARD APPLICATION Z
126z
124
16
Card JVM
122
12a
Card Operating System
Communicator
Integrated Circuit Card
209
Session Key
14
136
Terminal Application
134
Integrated Circuit Card Driver
132
Communicator Driver
12b
Terminal Communicator
Terminal

FIGURE 13

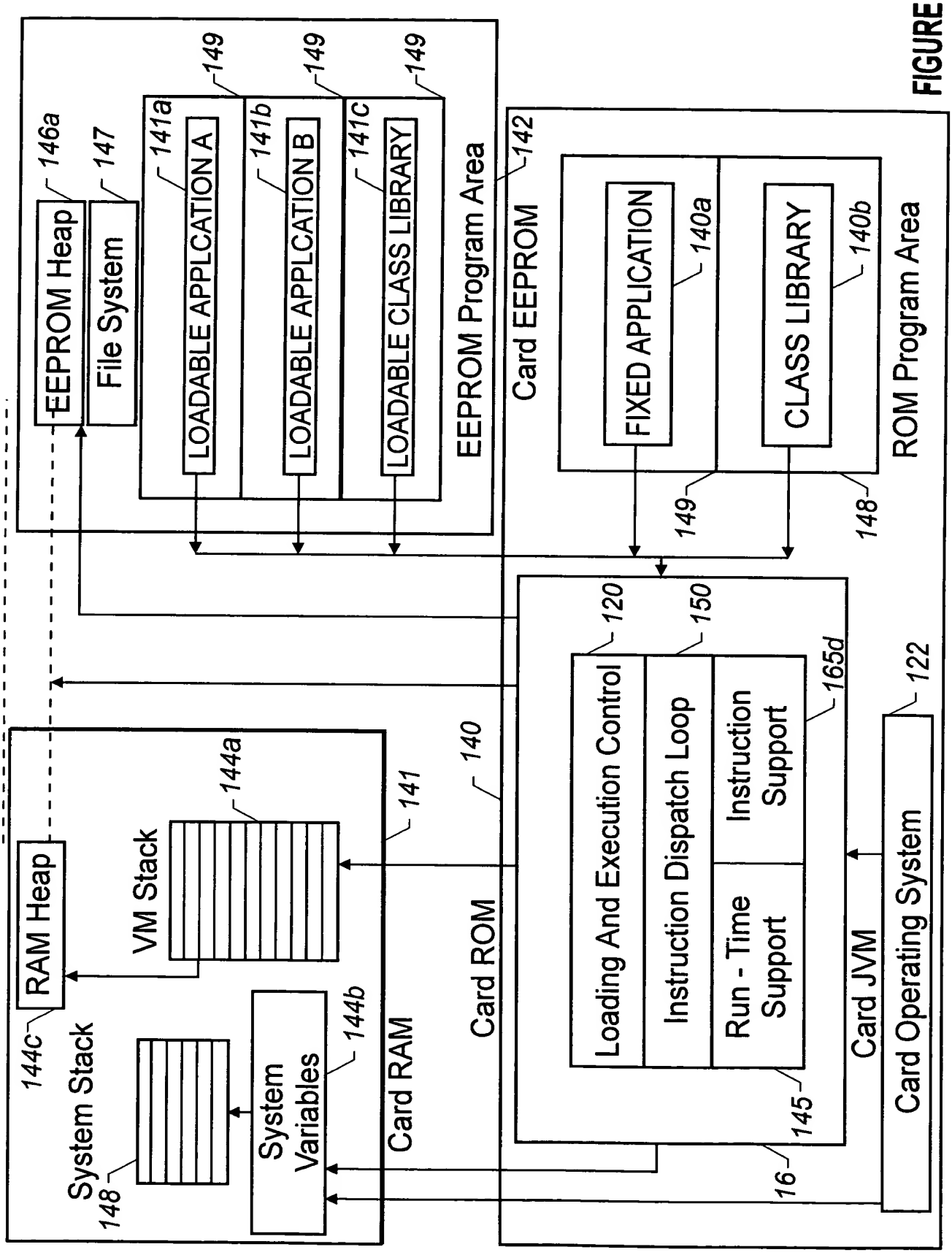

EEPROM Heap
146a
File System
147
LOADABLE APPLCATION A
141a
LOADABLE APPLCATION B
141b
LOADABLE CLASS LIBRARY
141c
149
EEPROM Program Area
142
Card EEPROM
FIXED APPLICATION
140a
CLASS LIBRARY
140b
ROM Program Area
148
RAM Heap
144c
VM Stack
144a
System Stack
148
System Variables
144b
Card RAM
141
Card ROM
140
Loading And Execution Control
120
Instruction Dispatch Loop
150
Run - Time Support
Instruction Support
165d
145
Card JVM
16
Card Operating System
122

FIGURE 14

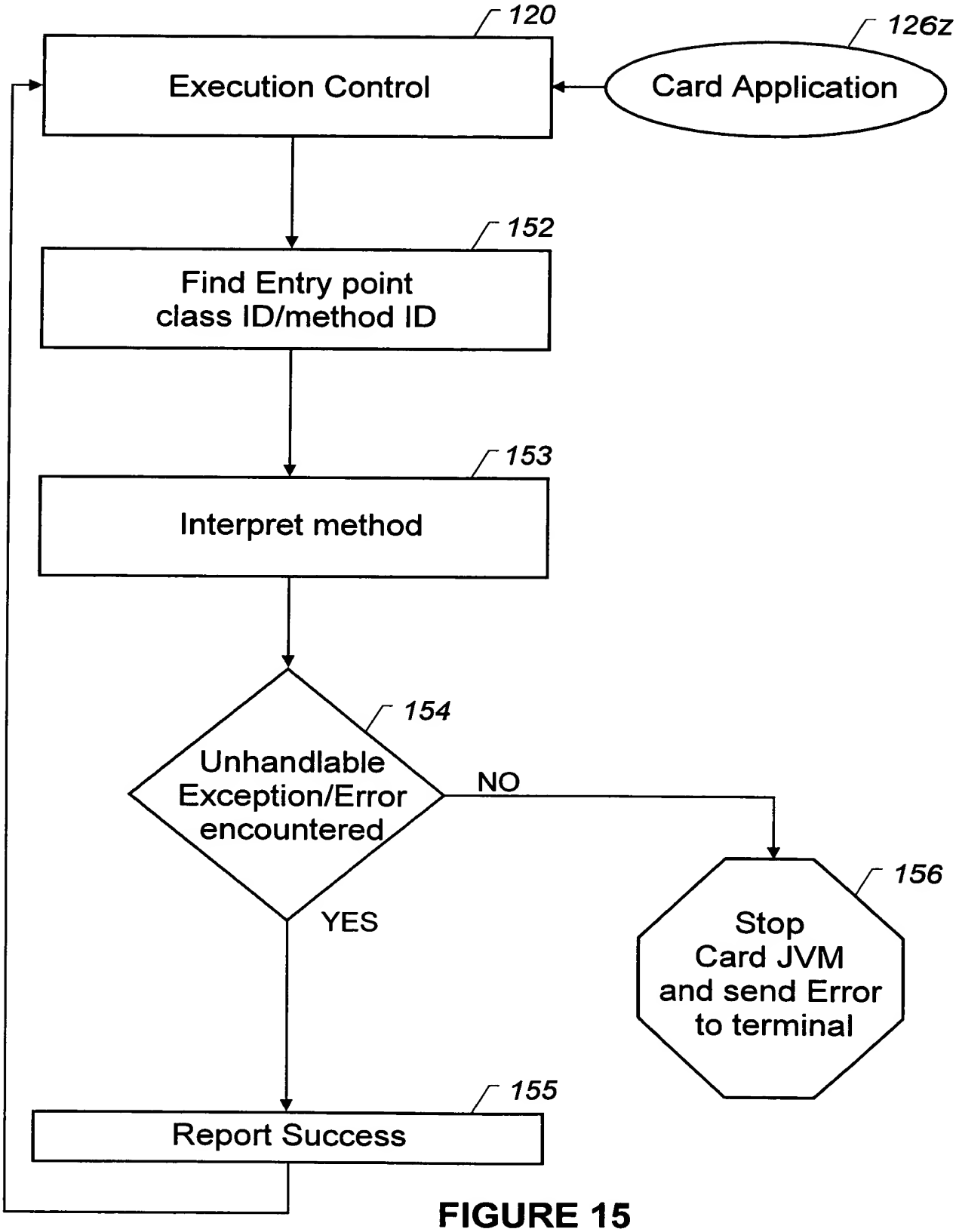

120
Execution Control
126z
Card Application
152
Find Entry point
class ID/method ID
153
Interpret method
154
Unhandlable
Exception/Error
encountered
NO
YES
156
Stop
Card JVM
and send Error
to terminal
155
Report Success

FIGURE 15

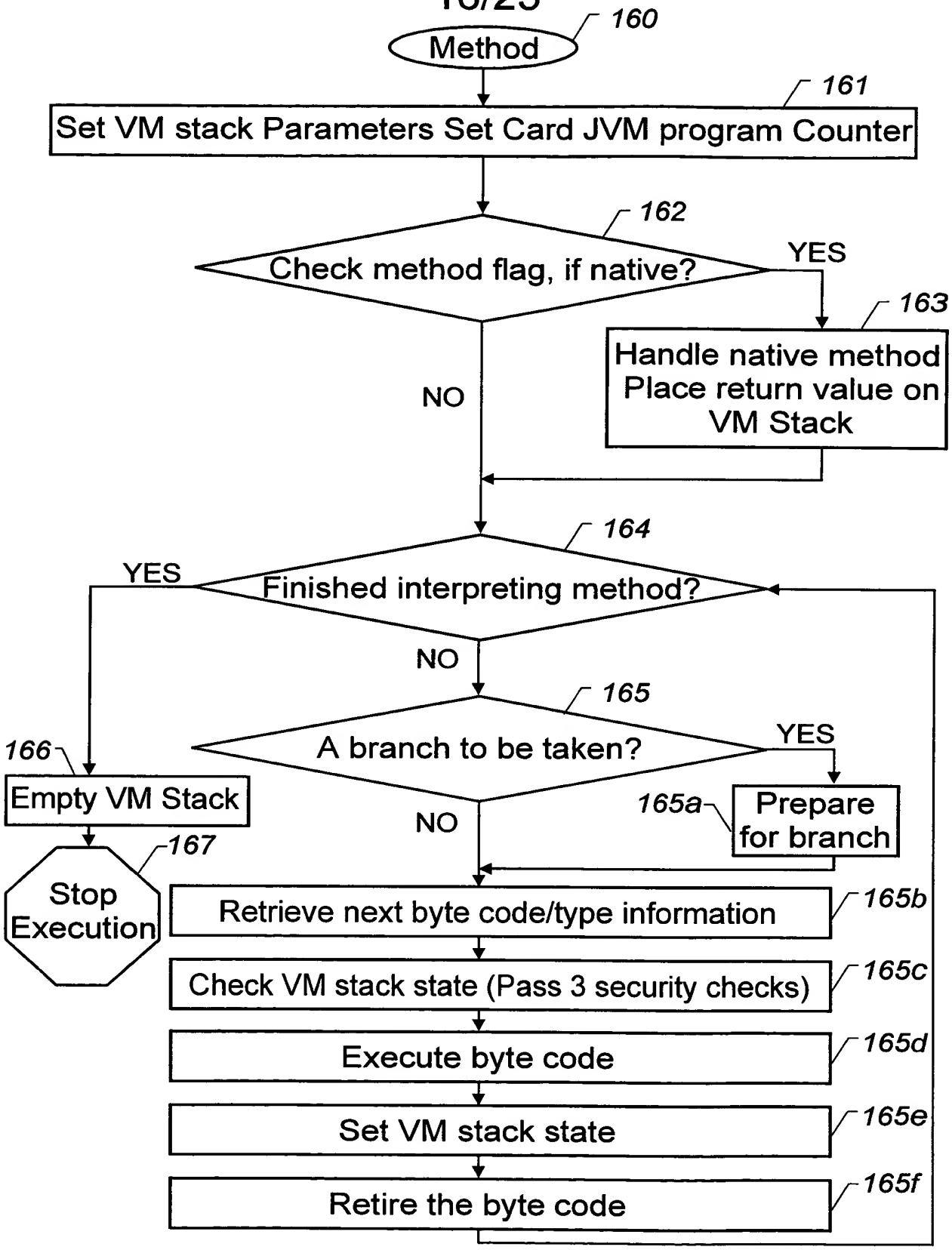

160
Method
161
Set VM stack Parameters Set Card JVM program Counter
162
Check method flag, if native?
YES
163
Handle native method
Place return value on
VM Stack
NO
164
YES
Finished interpreting method?
NO
165
A branch to be taken?
YES
166
Empty VM Stack
165a
Prepare
for branch
NO
167
Stop
Execution
Retrieve next byte code/type information
165b
Check VM stack state (Pass 3 security checks)
165c
Execute byte code
165d
Set VM stack state
165e
Retire the byte code
165f

FIGURE 16

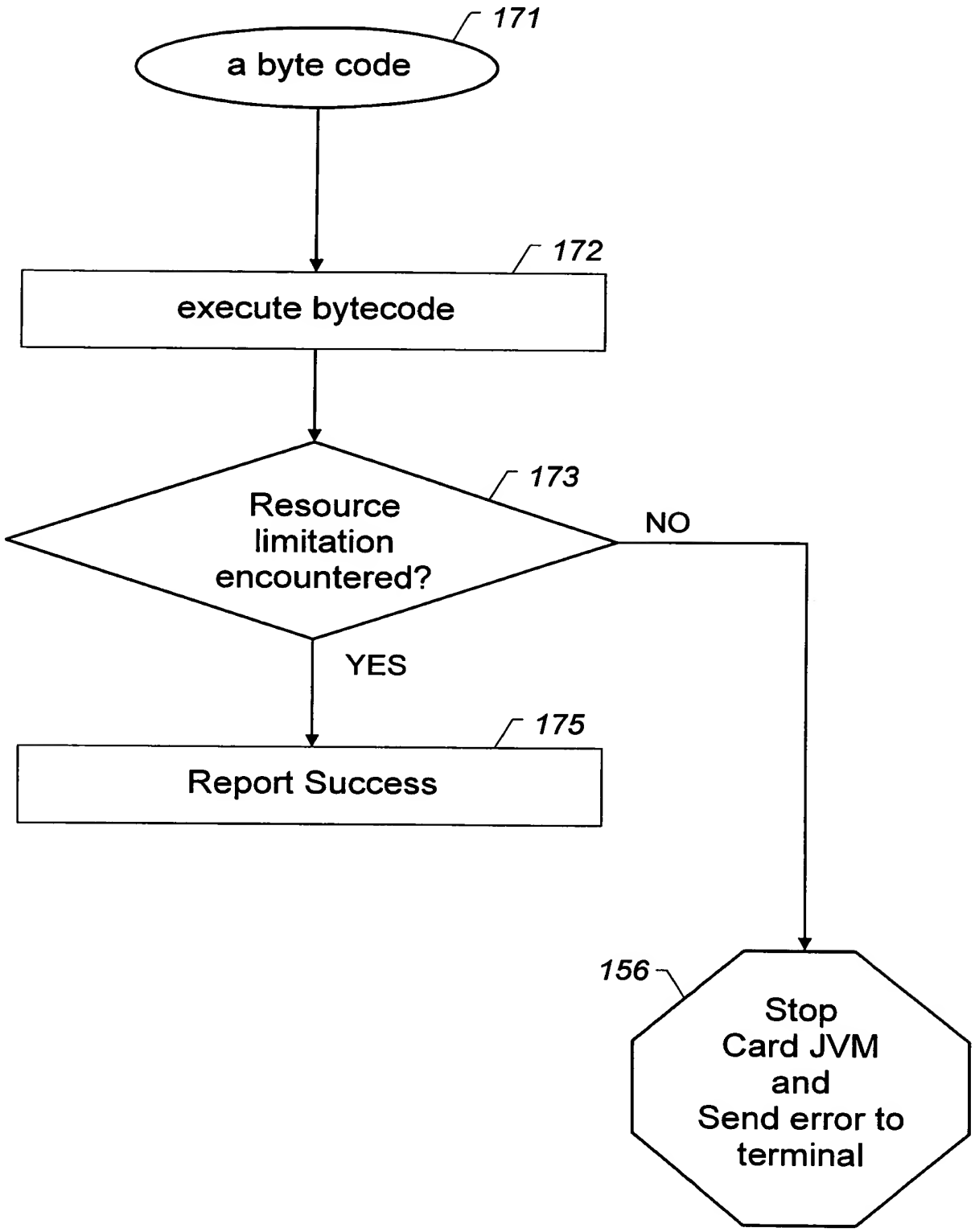

171
a byte code
172
execute bytecode
173
Resource limitation encountered?
NO
YES
175
Report Success
156
Stop Card JVM and Send error to terminal

FIGURE 17

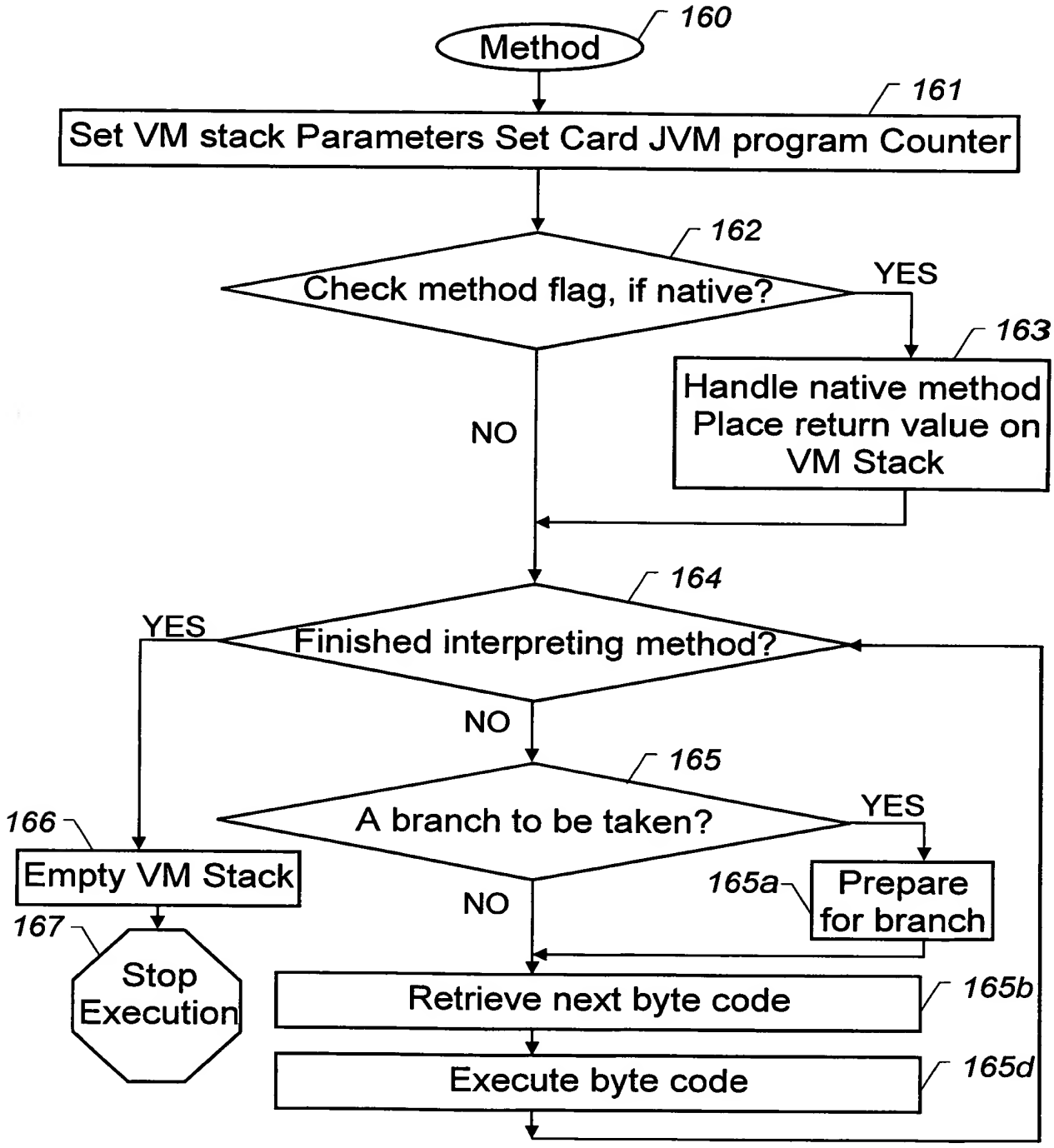

Method
160
Set VM stack Parameters Set Card JVM program Counter
161
Check method flag, if native?
162
YES
Handle native method Place return value on VM Stack
163
NO
Finished interpreting method?
164
YES
NO
A branch to be taken?
165
YES
NO
Prepare for branch
165a
Empty VM Stack
166
Stop Execution
167
Retrieve next byte code
165b
Execute byte code
165d

FIGURE 18

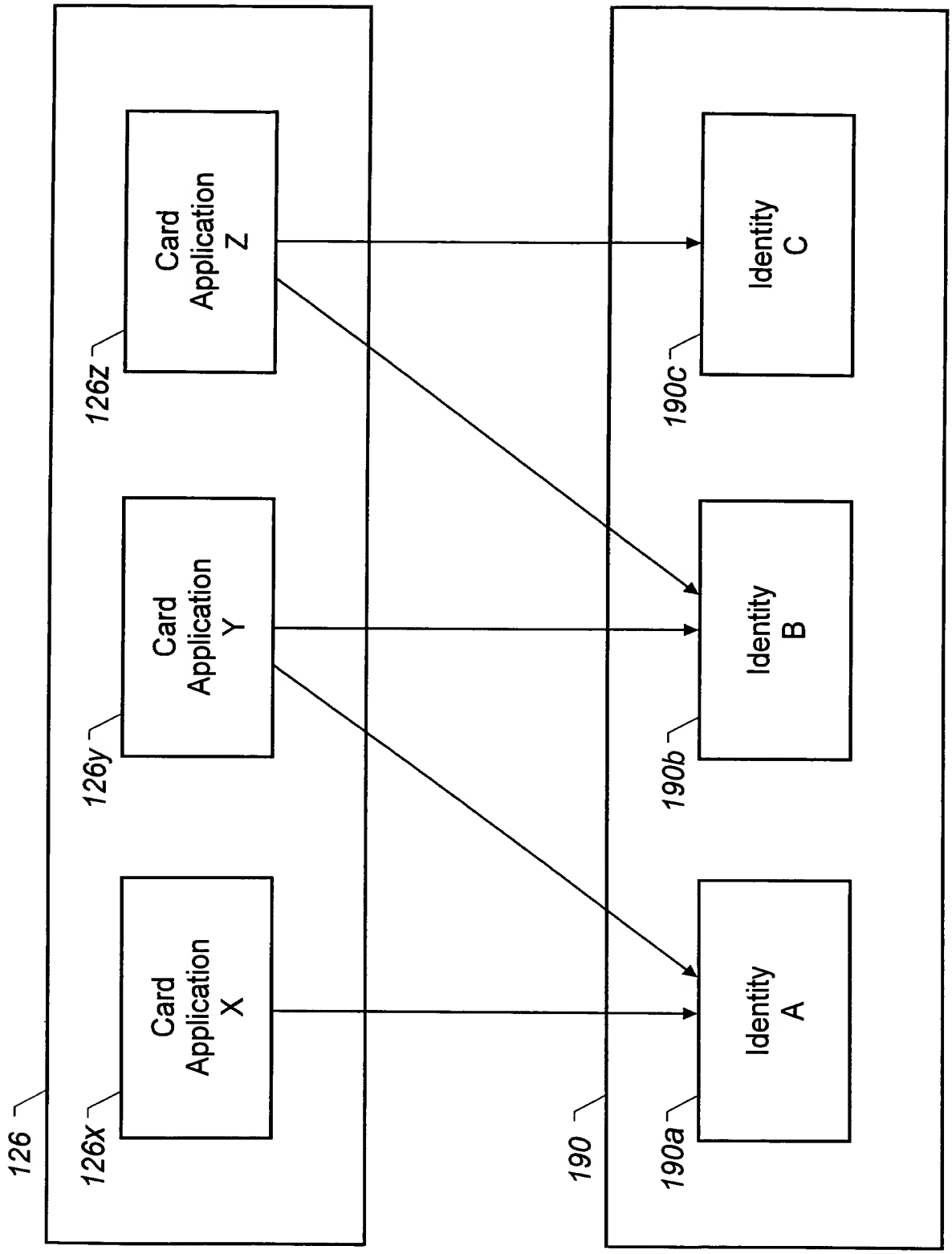

126
126x
Card Application X
126y
Card Application Y
126z
Card Application Z
190
190a
Identity A
190b
Identity B
190c
Identity C

FIGURE 19

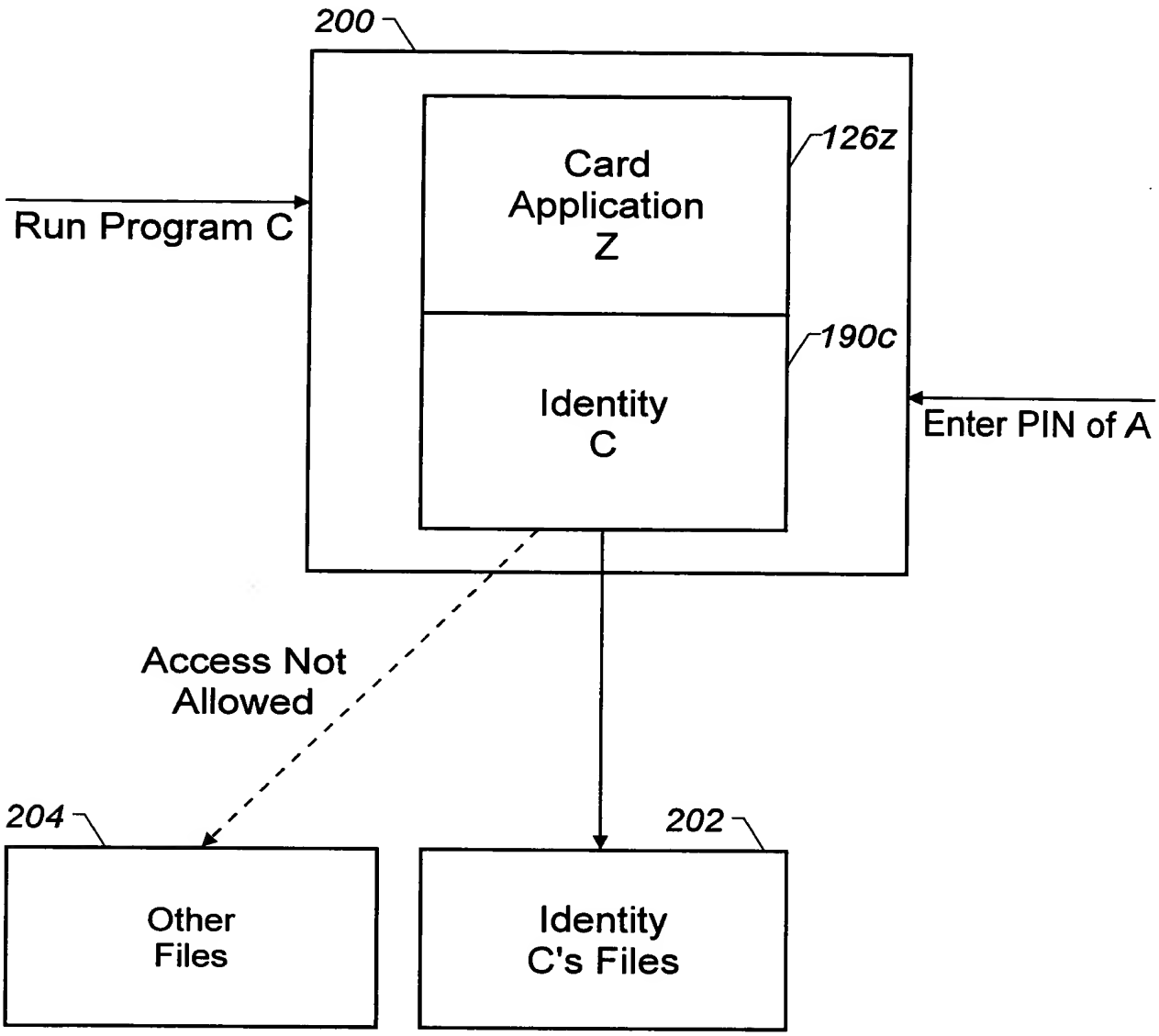

200
126z
Card Application Z
190c
Identity C
Run Program C
Enter PIN of A
Access Not Allowed
204
202
Other Files
Identity C's Files

FIGURE 20

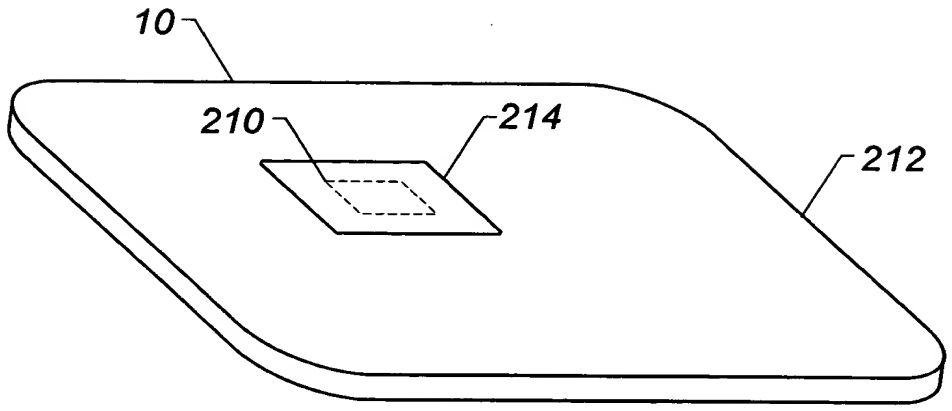


FIGURE 21

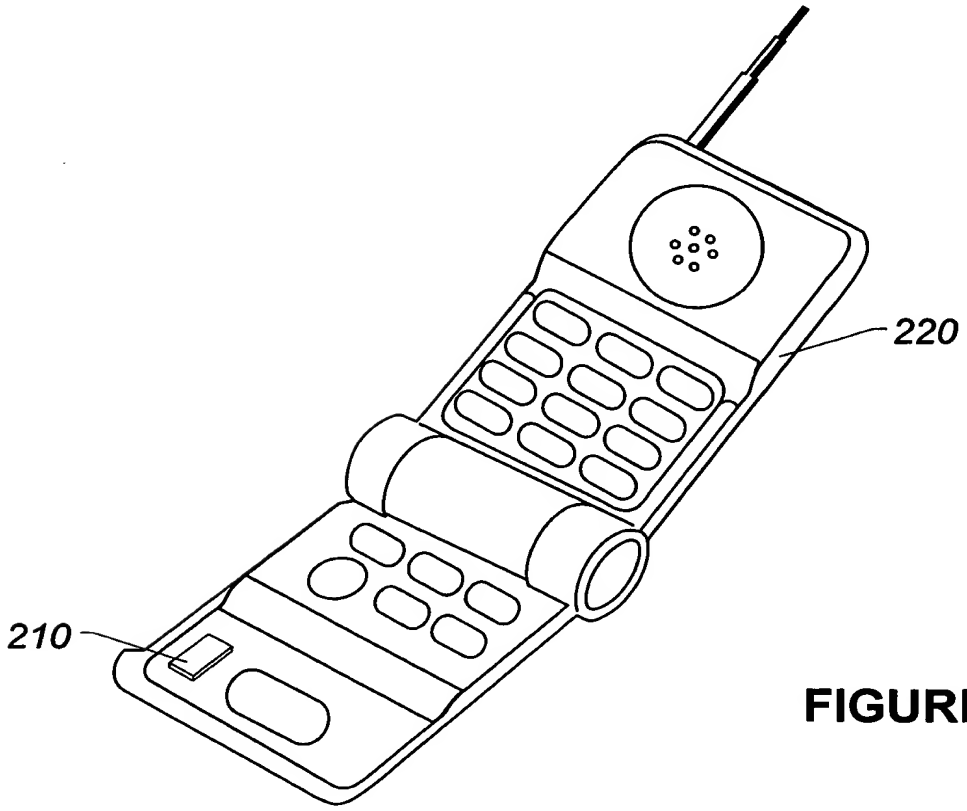


FIGURE 22

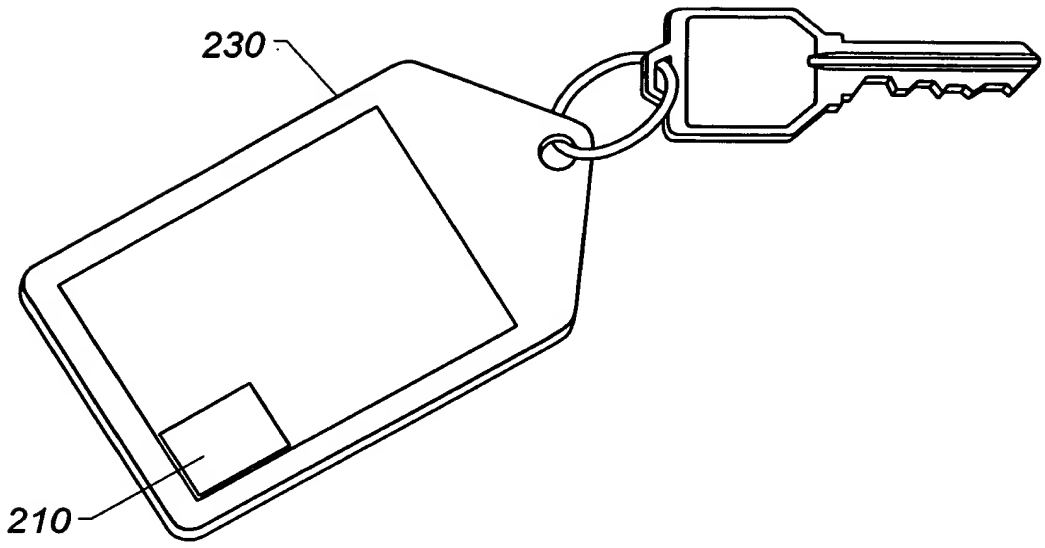


FIGURE 23

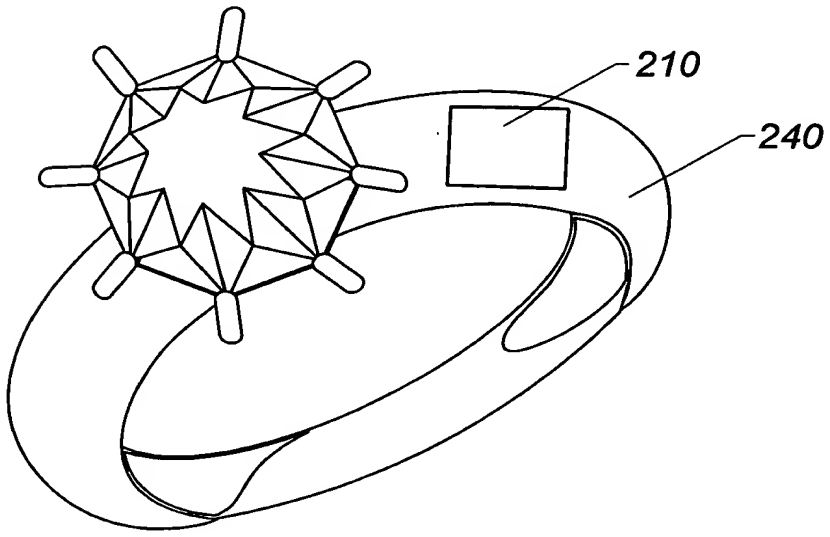


FIGURE 24

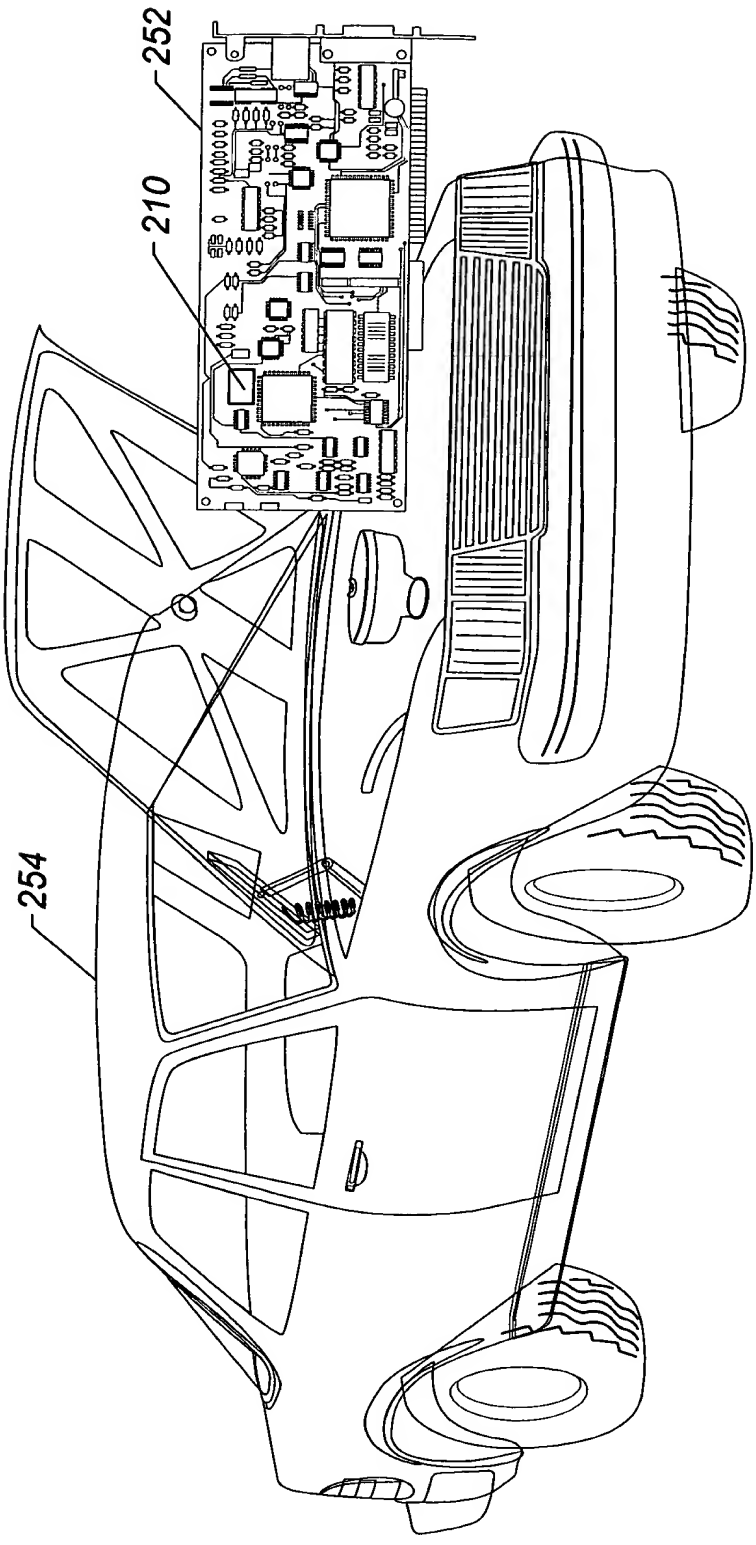


FIGURE 25